

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Možnosti hlasového ovládání bezpilotních dronů

Drone Motion Control Based on Voice Commands

Zadání bakalářské práce

Student:

Jan Vařák

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R059 Mobilní technologie

Téma:

**Možnosti hlasového ovládání bezpilotních dronů
Drone Motion Control Based on Voice Commands**

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem bakalářské práce je navrhnout komunikační schéma mezi uživatelem a bezpilotním dronem s možností hlasového ovládání samotného stroje, či jeho příslušenství. Cílem je definovat jednotlivé hlasové příkazy a popsat algoritmus detekce a přiřazení správného kroku k hlasovému vstupu.

Zadání:

1. Detailně nastudujte problematiku hlasového ovládání bezpilotních modelů.
2. Proveďte rešerši stávajících řešení pro hlasové ovládání bezpilotních modelů s využitím mobilních zařízení.
3. Navrhněte komunikační schéma mezi mobilním zařízením a modelem včetně definice možných technologií a protokolů.
4. Vytvořte teoretický návrh hlasových příkazů nutných pro hlasové ovládání modelu.
5. Definujte nutné předpoklady pro možnou praktickou implementaci včetně specifikace možných problémů vycházejících z vlastního návrhu.

Seznam doporučené odborné literatury:

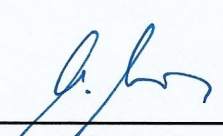
1. Daniel Eggert, Communication with the Quadcopter, 2014.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Filip Řezáč, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017


doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2017


.....

Rád bych na tomto místě poděkoval všem, kteří mi pomohli s touto bakalářskou prací a hlavně vedoucímu mé bakalářské práce Ing. Filipu Řezáčovi, Ph.D. za jeho rady a připomínky.

Abstrakt

Cílem bakalářské práce je navrhnout komunikační schéma mezi uživatelem a bezpilotním dronem s možností hlasového ovládání samotného stroje či jeho příslušenství. Toto schéma je vyhotoveno na základě metod pro zpracování řečového signálu a metod pro rozpoznávání slov, které jsou v práci popsány. Dále jsou zde definovány jednotlivé hlasové příkazy a popsán algoritmus detekce a přiřazení správného příkazu k hlasovému vstupu. V neposlední řadě práce také obsahuje rešerši stávajících řešení pro hlasové ovládání bezpilotních modelů.

Klíčová slova: dron, DTW, gimbal, hlasové ovládání, metody krátkodobé analýzy, rozpoznávání hlasu, skrytý Markovův model, zpracování řečového signálu

Abstract

The aim of this Bachelor's thesis is to design a communication scheme between user and unmanned drone with possibility of voice control of the machine itself or its parts. This scheme is made on the basis of methods for speech signal processing and methods for word recognition, which are described in this thesis. In this work, there are also defined particular voice commands and an algorithm for detection and assigning proper commands to voice input. Last but not least, this work also contains research of existing solutions for voice control of unmanned vehicles.

Key Words: drone, DTW, gimbal, hidden Markov model, methods of short-time analysis, speech signal processing, speech recognition, voice control

Obsah

| | |
|--|-----------|
| Seznam použitých zkratk a symbolů | 7 |
| Seznam obrázků | 8 |
| Seznam tabulek | 9 |
| 1 Úvod | 10 |
| 2 Problematika hlasového ovládání bezpilotních modelů | 11 |
| 2.1 Dron | 11 |
| 2.2 Historie dronů | 12 |
| 2.3 Princip létání dronů | 12 |
| 2.4 Ovládání dronů | 13 |
| 2.5 Rešerše stávajících řešení pro hlasové ovládání bezpilotních modelů | 15 |
| 3 Komunikační schéma mezi mobilním zařízením a modelem | 19 |
| 3.1 Protokoly pro ovládání dronů | 19 |
| 3.2 Formát zprávy pro přenos informací | 19 |
| 3.3 Zpracování řečového signálu | 21 |
| 3.4 Metody krátkodobé analýzy | 24 |
| 3.5 Rozpoznávání řeči | 28 |
| 3.6 Dynamické borcení časové osy | 32 |
| 3.7 Další možnosti rozpoznávání řeči | 34 |
| 3.8 Návrh komunikačního schématu mezi mobilním zařízením a dronem | 34 |
| 4 Teoretický návrh hlasových příkazů | 37 |
| 4.1 Příkazy pro ovládání gimbalu | 38 |
| 4.2 Konkrétní příkazy dle způsobu využití dronu | 39 |
| 5 Předpoklady pro možnou praktickou implementaci a specifikace možných problémů | 42 |
| 6 Závěr | 43 |
| 7 Seznam použité literatury | 44 |
| Literatura | 44 |

Seznam použitých zkratek a symbolů

| | |
|---------|---|
| AWS | – Amazon Web Services |
| DTW | – Dynamic Time Warping(Dynamické borcení časové osy) |
| GPS | – Global Positioning System(Globální polohový systém) |
| IoT | – Internet of Things(Internet věcí) |
| MAVLink | – Micro Air Vehicle Link |
| MFCC | – Mel-frequency cepstral coefficients(Melfrekvenční keprální koeficienty) |
| PC | – Personal Computer(Osobní počítač) |
| PCM | – Pulse-code modulation(Pulzně kódová modulace) |
| TCP | – Transmission Control Protocol |
| UAV | – Unmanned Aerial Vehicle(Bezpilotní etoun) |
| UDP | – User Datagram Protocol |
| USB | – Universal Serial Bus(Univerzální sériová sběrnice) |
| Wi-Fi | – Wireless Fidelity |

Seznam obrázků

| | | |
|----|---|----|
| 1 | Dron DJI Phantom | 11 |
| 2 | Princip létání dronů[3]. | 12 |
| 3 | Aplikace freeflight 3 pro ovládání dronů společnosti Parrot. | 14 |
| 4 | Kanály pro ovládání dronu[4]. | 14 |
| 5 | Dron Fineco FX-4V | 15 |
| 6 | Architektura hlasově ovládaného dronu IRIS+[5]. | 16 |
| 7 | Úryvek kódu pro sestavení textového řetězce ze sériového toku[9]. | 17 |
| 8 | Signál slova dron a počet průchodů nulou v něm. | 25 |
| 9 | Signál slova dron a průběh jeho krátkodobé energie. | 26 |
| 10 | Jednoduché schéma rozpoznávání řeči. | 29 |
| 11 | Stavový diagram Markovova řetězce. | 29 |
| 12 | Levo-pravý Markovův model o šesti stavech[18]. | 31 |
| 13 | Naznačená cesta v matici G. | 33 |
| 14 | Návrh komunikačního schématu mezi mobilním zařízením a dronem. | 35 |
| 15 | Diagram aktivit pro rozpoznávání řeči. | 36 |
| 16 | Gimbal Mini 3D Pro Feiyu Tech | 37 |

Seznam tabulek

| | | |
|---|--|----|
| 1 | Protokoly a porty pro zasílání příkazů do dronu AR.Drone 2.0[13]. | 19 |
| 2 | Nejčastěji používané hodnoty pro ovládání dronu pomocí AT příkazů[14]. | 20 |

1 Úvod

V dnešní době jsou drony stále více používány nejen pro záznam videa, ale také například v zemědělství, energetice nebo pro účely integrovaného záchranného systému. Jelikož je dnes kladen stále větší důraz na automatizaci a zefektivňování nejrůznějších procesů, jsou tyto nároky kladeny i na ovládání dronů. Mimo jiné, toho lze dosáhnout s pomocí zavedení hlasového ovládání dronů, kterým se tato práce zabývá. Na trhu se momentálně objevují modely s funkčním hlasovým ovládáním, jedná se však převážně o hobby drony vhodné především pro zábavu. Avšak hlasové ovládání celého dronu naráží na určité problémy, jež jsou vysvětleny v této práci, a tak se jako vhodnější řešení pro profesionální účely nabízí zavést hlasové ovládání pouze pro periodické funkce a periferie dronu.

Dobrým příkladem periferie může být gimbal, známý také jako Kardanův závěs, což je zařízení umožňující zavěšenému přístroji volný pohyb po třech osách, přičemž tímto zavěšeným přístrojem obvykle bývá kamera. Právě hlasové ovládání gimbalu a na něm zavěšené kamery je žádoucí, protože v praxi jsou pro tyto účely potřební dva lidé – jeden člověk pro obsluhu dronu a druhý pro obsluhu kamery. Pokud však zavedeme hlasové ovládání, mohou být obě funkce vykonávány jedním člověkem, což je jednak levnější, a navíc se zavedením hlasových příkazů může spousta složitých úkonů zjednodušit.

Tato práce se nejprve zabývá problematikou hlasového ovládání bezpilotních modelů. Dále obsahuje rešerši stávajících řešení pro hlasové ovládání bezpilotních modelů. V další kapitole je popsáno, jakým způsobem je možné s drony komunikovat, a také jsou zde zmíněny metody zpracování řečového signálu a rozpoznávání řeči. Následující kapitola se zabývá návrhem hlasových příkazů pro ovládání gimbalu na základě konkrétních využití dronů. Poslední kapitola pojednává o předpokladech pro možnou praktickou implementaci a o specifikaci možných problémů.

2 Problematika hlasového ovládání bezpilotních modelů

2.1 Dron

Bezpilotní letadla můžeme někdy označovat jako UAV (z anglického Unmanned Aerial Vehicle) nebo také dron (z anglického drone). Dron je letadlo bez posádky, které může být řízeno na dálku nebo může létat samostatně pomocí předprogramovaných letových plánů či pomocí složitějších dynamických autonomních systémů. Bepilotní letadla se často používají v armádě k průzkumným i útočným letům. Dále se používají také k mnoha civilním úkolům, například k hašení požárů, policejnímu sledování nebo průzkumu terénu.

Navzdory tomu, že postupně začínají převažovat multifunkční letadla, řadíme bezpilotní letadla zpravidla do následujících funkčních kategorií:

1. **Cíle a návnady** poskytují pozemní a vzdušné cíle simulující nepřátelská letadla nebo střely
2. **Průzkumná** poskytují informace o bojišti
3. **Bojová** schopné útoku ve velmi rizikových misích
4. **Logistická** drony speciálně navržené pro logistické účely
5. **Výzkumná a vývojová** používané k dalšímu vývoji UAV technologií
6. **Civilní a komerční** drony speciálně navržené pro civilní nebo komerční aplikace

V této práci se budu zabývat drony z poslední zmíněné kategorie[1].

Pro ilustraci uvádím obrázek¹ dronu níže:



Obrázek 1: Dron DJI Phantom

¹<http://www.dji.com/phantom>

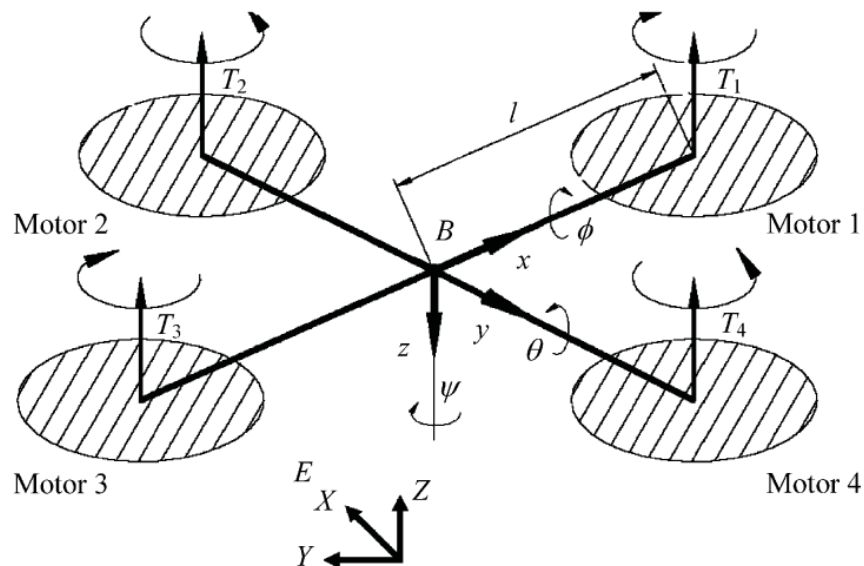
2.2 Historie dronů

Historie dronů sahá až ke známému americkému vynálezci srbského původu jménem Nikola Tesla, který si nechal v roce 1898 patentovat tzv. teleautomatizaci, což představovalo dálkové ovládání motorové loďky na vodě. Později se ukázalo, že v jeho poznámkách byly i úvahy o sestrojení bezpilotního leteckého systému.

Ještě dříve však existovaly horkovzdušné balóny bez pilotů, které již v roce 1849 shazovaly výbušniny při útoku Rakousko-Uherska na zabarikádované nepřátele v Benátkách. První bezpilotní letadlo vyprojektoval v roce 1916 anglický inženýr Archibald Montgomery Low, známý vynálezce řízených raket, torpéd apod. Toto letadlo se jmenovalo Aerial Target (Vzdušný cíl). Ihned následovala spousta letadel řízených na dálku, jejichž hlavním účelem bylo sloužit jako dálkově ovládaná torpéda, např. experimentální bezpilotní letadlo Kettering Bug, které bylo schopné zasáhnout cíl až na 64 kilometrů a bylo poprvé úspěšně testováno na konci roku 1918[2].

2.3 Princip létání dronů

Princip létání dronů vysvětlím na kvadrokoptéře, což je, jak už sám název napovídá, dron se čtyřmi vrtulemi, přičemž jsou tyto vrtule v jedné rovině. Kvadrokoptéra má 6 stupňů volnosti (pohybuje se po třech osách a může kolem tří os i rotovat), avšak pouze 4 vrtule. Kvůli vyrovňování se dvě z vrtulí točí po směru hodinových ručiček a další dvě naopak v protisměru hodinových ručiček, tyto vrtule jsou pak umístěny diagonálně (tzv. do kříže) naproti sebe.



Obrázek 2: Princip létání dronů[3].

Pohyby kvadrokoptéry jsou řízeny změnou otáček vrtulí, což má za příčinu změnu úhlové rychlosti vrtulí.

Pro změnu výšky letu je třeba snížit, případně zvýšit otáčky všech vrtulí o stejnou hodnotu; podle toho, jestli má dron stoupat nebo klesat.

Let dopředu či dozadu je řízen snížením otáček předního nebo zadního páru vrtulí o stejnou hodnotu.

Let doleva nebo doprava je řízen snížením otáček pravých či levých vrtulí o stejnou hodnotu. Otočení dronu docílíme změnou otáček všech čtyř vrtulí. Změny při rotaci jsou následující: jedna dvojice vrtulí své otáčky sníží o určitou hodnotu, přičemž zbylá dvojice své otáčky o stejnou hodnotu naopak zvýší[3].

2.4 Ovládání dronů

Nyní již víme, jakým způsobem se dron pohybuje, a můžeme si vysvětlit, jak přesně probíhá ovládání dronu.

Ovládání se skládá ze dvou částí: vysílače, který se nachází v ovládacím zařízení, a přijímače, který je součástí ovládaného dronu. Vysílač vysílá řídicí signál, kterým se řídí servomotory dronu. Pro ovládání se využívá rádiového signálu.

Létající rádiově ovládané modely byly ovládány pomocí rádiových vln o frekvenci 35 MHz, nicméně tento způsob se již nepoužívá. Dnešní profesionální drony komunikují nejčastěji prostřednictvím rádiových vln o frekvenci 2,4 GHz. Využívají tedy standardů IEEE 802.11. Jako přijímač a vysílač se používají počítače, které jsou vzájemně propojeny bezdrátovou sítí. Zvýšení frekvence přináší větší odolnost proti rušení od dalších vysílačů, protože si vysílač automaticky najde jiný kanál, pokud se v okolí nachází jiný vysílač pracující s daným kanálem nebo si soupravy několikrát za sekundu tyto kanály mění, čímž předcházejí rušení. Negativní vlastností vyšší frekvence je menší prostupnost pevnými objekty.

2.4.1 Zařízení pro ovládání dronů

V dnešní době jsou využívány hlavně dva druhy ovládacích zařízení. Prvním z nich je mobilní zařízení, kde ovládání probíhá pomocí aplikace a propojení dronu a mobilního zařízení prostřednictvím Wi-Fi.

Aplikace běžně umožňují sledovat přenos videa, polohu dronu pomocí GPS, rychlost letu, stav baterie a další informace. Samotné ovládání probíhá buď pomocí softwarových tlačítek, nebo gyroskopu v mobilním zařízení. Na obrázku² níže můžeme vidět aplikaci využívající softwarových tlačítek.

Druhým typem zařízení je ovladač obsahující 2 páky a případně i tlačítka pro speciální funkce nebo kalibraci. V dnešní době je součástí ovladače často i displej poskytující nejružnější informace o dronu a přenos videa z kamery dronu.

²https://community.parrot.com/t5/tkb/allarticlesprintpage/tkb-id/Bebop_2@tkb



Obrázek 3: Aplikace freeflight 3 pro ovládání dronů společnosti Parrot.

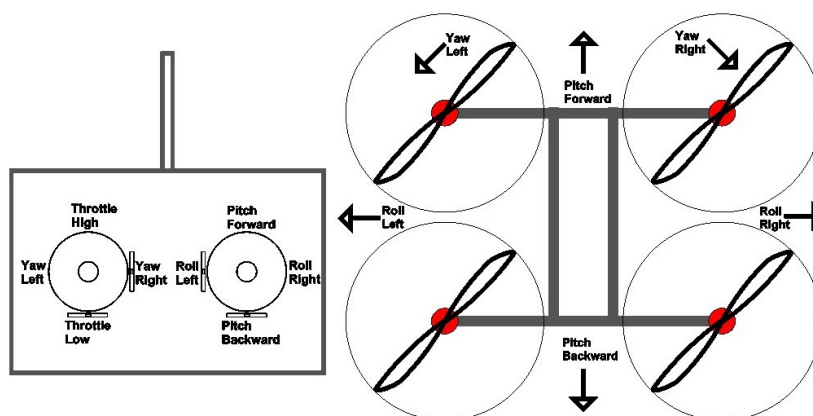
Ovládání probíhá pomocí kanálů. Pro kvadrokoptéry se používá ovládání prostřednictvím čtyř kanálů, kterými jsou Yaw, Pitch, Roll a Throttle.

Roll se používá pro pohyb do stran a probíhá po posunutí pravé páky doleva nebo doprava.

Pitch se používá pro pohyb dopředu nebo dozadu dle pohybu pravé páky.

Yaw se používá pro rotaci dronu po nebo proti směru hodinových ručiček pomocí posunutí levé páky doleva či doprava.

Throttle slouží pro ovládání stoupání a klesání dronu. Pro stoupání se posune levá páka dopředu a pro klesání naopak dozadu[4].



Obrázek 4: Kanály pro ovládání dronu[4].

2.5 Rešerše stávajících řešení pro hlasové ovládání bezpilotních modelů

Jak už bylo popsáno v úvodu, drony ovládané hlasem již existují, ale jedná se převážně o hobby drony určené pro zábavu. U hlasového ovládání dronu totiž nastává problém poměrně velké odezvy na hlasové příkazy a s tím spjatou legislativní stránkou věci, kdy dron ovládaný hlasem může být rizikem při nečekaných situacích, které by vyžadovaly rychlou reakci pilota. Proto se hlasové ovládání celého dronu u profesionálních dronů nepoužívá. Nicméně existuje i několik projektů zabývajících se hlasovým ovládáním, které budou společně s příkladem hobby dronu ovládaného hlasem, zmíněny níže.

2.5.1 Hlasem ovládaný hobby dron Fineco FX-4V

Hlasové ovládání tohoto dronu je prováděno přes mikrofon a obsahuje 21 příkazů. Mezi nimi můžeme najít základní funkce pro ovládání letu: take off, go up, go down, go forward, go back, go left, go right, turn left, turn right, U turn, auto pilot, come back, cancel. Dále pak příkazy pro obraty a přednastavené show: dancing, flip forward, flip back, flip left, flip right, stunt one, stunt two a stunt three.

Obrázek³ tohoto dronu je uveden níže:



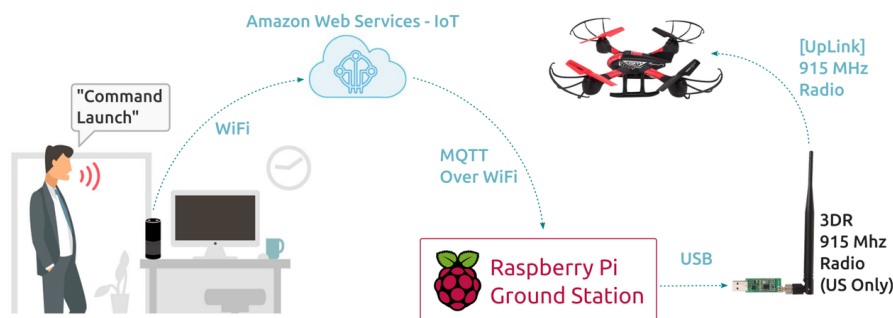
Obrázek 5: Dron Fineco FX-4V

2.5.2 Hlasem ovládaný dron IRIS+

Tento projekt využívá Raspberry Pi 2 model B a Amazon Alexa Echo, což je zařízení určené k hlasovému ovládání domácích spotřebičů. Softwarová část je zpracována pomocí Amazon Web Services AWS IoT, což je platforma pro komunikaci se zařízeními v internetu věcí. Amazon Alexa Echo zpracovává hlasové příkazy, které jsou následně přenášeny jako textové řetězce do AWS IoT. Odtud se příkazy pomocí komunikačního standardu MQTT přenášejí do Raspberry Pi, které funguje jako prostředník mezi AWS IoT a dronem. Jako vysílač se používá rádiový modul 3DR, který se do Raspberry Pi zapojí pomocí USB portu a pracuje na frekvenci 915

³http://www.finecosz.com/products_detail/productId=68.html

MHz. Vysílač zasílá již zpracované hlasové příkazy do dronu ve formě příkazů, kterým dron rozumí. Na obrázku níže je vyobrazena architektura tohoto řešení. Hlavními nevýhodami tohoto řešení jsou poměrně dlouhá prodleva mezi vyslovením příkazu a finálním vykonáním příkazu dronem a nutnost připojení Amazon Alexa Echo a Raspberry Pi k Internetu, což je při práci v terénu nevhodné. Problém připojení je možné částečně vyřešit pomocí funkce Device Shadows v AWS IoT, která využívá poslední známé konfigurace, pokud je zařízení offline[5].



Obrázek 6: Architektura hlasově ovládaného dronu IRIS+[5].

2.5.3 Hlasem ovládaný Bebop dron

Tento projekt realizuje společnost Mobiquity Inc. a je podobný výše popsanému projektu. I zde se využívá IoT platformy od společnosti Amazon AWS IoT. Je zde ovšem využito jiné zařízení pro vstup hlasových příkazů, a sice zařízení Amazon Echo Dot. Dron Parrot Bebop, které je ovládáno pomocí příkazů zasílaných z Raspberry Pi pomocí knihovny node.JS a knihovny node-bebop[6]. Tento projekt částečně řeší imobilitu, vzhledem k potřebě být připojen do Wi-Fi sítě, za pomoci linuxového démona SystemD. Ten vytvoří dva síťové stavy. První stav je Raspberry Pi, vystupující jako hotspot pro konfiguraci známých sítí v případě nenalezení známých sítí. Druhý stav pak je, že se Raspberry Pi připojí na dvě různé Wi-Fi sítě zajišťující Internetovou konektivitu[7].

2.5.4 Parrot AR.Drone 2 ovládaný za použití Intel RealSense SDK

Tento projekt je vyvíjen dvojicí Marco Dal Pino a Marco Minerva. Jeho základem je komunikace s dronem AR.Drone 2 pomocí knihovny psané v programovacím jazyce C#. Pro rozpoznávání hlasu a převod na hlasové příkazy je použito programové vybavení od společnosti Intel RealSense SDK. Dronu jsou zasílány příkazy v podobě AT příkazů, které umožňují základní operace s dronem, jako například vzlet, přistání a létání v různých směrech. Pomocí AT příkazů můžeme také získat stream videa v reálném čase. Jelikož si dron AR.Drone 2 vytváří svou Wi-Fi síť, je nutné se na tuto síť připojit a dále posílat požadované příkazy z PC či z telefonu na konkrétní IP adresu. RealSense SDK pak poskytuje relativně jednoduchý přístup k rozpoznávání hlasu za

pomoci předem definovaného seznamu klíčových slov. Pro tyto účely bylo vytvořeno pole řetězců s jednoduchými příkazy: Takeoff, Land, Rotate Left, Rotate Right, Advance, Back, Up, Down, Left, Right, Stop, Dance. Tento přístup k hlasovému ovládaní je přímočařejší a praktičtější než dva výše zmíněné projekty. Není zde zapotřebí žádná interakce se servery přístupné skrze internet, tím pádem se jedná o vhodný projekt i do terénu[8].

2.5.5 Dron ovládaný hlasem založený na platformě Arduino

Tento projekt se od ostatních zmiňovaných liší tím, že není postaven na žádném komerčně dostupném dronu. Autor v článku popisuje i stavbu jednoduché trikoptéry. Cílem projektu je ovládat dron z mobilního telefonu přes bluetooth sériovou linku. Řídící část je vykonávána na mikropočítači Arduino Pro Mini s mikroprocesorem ATmega328. Část rozpoznávání hlasu probíhá na mobilním telefonu se systémem Android, který hlasové příkazy převádí na textové řetězce a ty pak zasílá na přes bluetooth do mikropočítače. Zde se postupným přičítáním posílaných znaků vytvoří požadovaný řetězec s příkazem, jak je předvedeno na obrázku níže.

```
void loop() {  
char c = Serial.read(); //Čtení znaků ze sériové linky.  
if (c == '#') {  
break; //Pokud slovo končí znakem # je smyčka přerušena.  
}  
voice += c; //Přičítání znaků pro vytvoření řetězce.  
}
```

Obrázek 7: Úryvek kódu pro sestavení textového řetězce ze sériového toku[9].

Řetězec voice z obrázku je pak porovnáván s příkazy známými pro Arduino a pomocí funkce analogWrite() je na motor, případně motory, zapsána analogová hodnota, která zapříčiní zvýšení či snížení tahu motoru či motorů. Tento projekt je určen zejména pro hobby účely, a to pro létání na krátké vzdálenosti. To proto, že je založen na technologii bluetooth. Zajímavostí je, že celý projekt, včetně stavby trikoptéry stál pouhých 11\$[9].

2.5.6 Hlasové ovládání periférií dronu Parrot Bebop 2

V předchozím textu byly uvedeny v současnosti dostupné projekty pro hlasové ovládání multi-koptér. Existuje však také projekt pro hlasové řízení samotných periférií s využitím dronu Parrot Bebop 2[10]. Tento dron je vybaven statickou kamerou v přední části těla. Pro snímání jednotlivých úhlů pohledu využívá výřezů z celkového snímaného obrazu, který je typu tzv. „rybí oko“. Tento projekt je realizován na Vysoké škole báňské – Technické univerzitě v Ostravě a pro interpretaci a zpracování hlasových příkazů byla vytvořena softwarová aplikace pro mobilní zařízení na operačním systému Android. Nástroj využívá hlasovou knihovnu CMU Sphinx a vývojové

prostředí pro drony Parrot – ARSDK, které dokáže detekovat správné klíčové slovo[11][12]. Následně aplikace vygeneruje sadu instrukcí pro kameru a ty jsou dále zaslány přes Wi-fi rozhraní do ručního ovladače tzv. SkyControlleru. Ten pouze zesílí a přepošle signál dále na přijímač samotného dronu, který provede zadaný příkaz. Časová odezva je v tomto případě velice krátká, jelikož způsob realizace využívá stávajících komunikačních protokolů a nahrazuje pouze ruční ovládání za hlasový vstup. Velkou výhodou je také lehká rozšiřitelnost na další sadu instrukcí a z principu tak realizovat i hlasové ovládání pohybu dronu.

3 Komunikační schéma mezi mobilním zařízením a modelem

Tato kapitola se zabývá celým průběhem komunikace s dronem. Vysvětlíme si, jak vlastně vypadají příkazy, kterými se drony ovládají, a jakým způsobem je do dronu posíláme. Dále jsou v této kapitole popsány možnosti zpracování řečového signálu a metody používané pro rozpoznávání řeči.

3.1 Protokoly pro ovládání dronů

Pro zasílání příkazů pro ovládání dronu se využívají protokoly transportní vrstvy referenčního modelu ISO/OSI. Zasílané příkazy a jejich struktura se mezi jednotlivými výrobci dronů mohou lišit, nicméně se dá říci, že jsou využívány protokoly UDP a TCP. Jako konkrétní příklad, k čemu se který protokol využívá a na jaký port jsou zasílána data, uvedu dron AR Drone 2.0.

| Protokol | Port | Využití |
|----------|------|--|
| UDP | 5554 | Informace o dronu, jako je pozice, rychlost, otáčky motoru, atd. jsou nazývány navdata a jsou dronem zasílány ovládacímu zařízení. |
| UDP | 5555 | Na tomto portu posílá AR Drone video do ovládacího zařízení. |
| UDP | 5556 | Dron je řízen pomocí AT příkazů, které jsou zasílány ovládacím zařízením. |
| TCP | 5559 | Tento port slouží jako "control port." Využívá se pro přenos kritických dat (proto se používá protokol TCP), jako jsou konfigurační data, a pro potvrzení (acknowledge) důležitých informací, jako je zasílání konfiguračních dat. |

Tabulka 1: Protokoly a porty pro zasílání příkazů do dronu AR.Drone 2.0[13].

3.2 Formát zprávy pro přenos informací

Formát zprávy, která je zasílána ovládacím zařízením dronu, se u různých výrobců a různých modelů dronu může lišit. Jako příklad zde opět uvádím AR Drone 2.0. Z předešlé kapitoly již víme, jaké protokoly jsou u AR Drone 2.0 využívány, a můžeme se podívat na to, jak vypadají příkazy, kterými je dron ovládán.

3.2.1 Skladba příkazu

Struktura příkazu pro ovládání vypadá následovně:

AT*REF=[Sequence number],[Flag bit-field],[Roll],[Pitch],[Gaz],[Yaw]<LF>

Vysvětlení:

Sekvenční číslo (Sequence number) je 1, pokud se jedná o první příkaz. Pokud ne, tak je sekvenční číslo rovno sekvenčnímu číslu předešlého příkazu +1.

Flag bit field při hodnotě 0 dron ignoruje hodnoty Roll, Pitch, Gaz a Yaw a vznáší se na místě. Při hodnotě 1 se dron pohybuje dle nastavení Roll, Pitch, Gaz a Yaw.

Roll se používá pro naklonění dronu doprava nebo doleva. Hodnota může nabývat -1 až 1, přičemž se jedná o číslo s pohyblivou řádovou čárkou.

Pitch se používá pro naklonění dronu dopředu nebo dozadu. Hodnota může nabývat -1 až 1, přičemž se jedná o číslo s pohyblivou řádovou čárkou.

Gaz vyjadřuje vertikální rychlost dronu. Hodnota může nabývat -1 až 1, přičemž se jedná o číslo s pohyblivou řádovou čárkou.

Yaw vyjadřuje úhlovou rychlost dronu. Hodnota může nabývat -1 až 1, přičemž se jedná o číslo s pohyblivou řádovou čárkou.

3.2.2 Příklad příkazu

Následující příkaz způsobí, že se dron bude otáčet ve směru hodinových ručiček při 75 procentech nastaveného výkonu.

AT* PCMD =1,1,0,0,0, 1061158912<LF>

Čísla pro roll, pitch, gaz a yaw se zadávají v podobě 32bitového decimálního čísla, které je převedeno na číslo hexadecimální, přičemž následným převodem hexadecimálního čísla na číslo s pohyblivou řádovou čárkou získáme skutečnou hodnotu. V tomto případě 0,75, protože decimální číslo 1061158912 je 3F400000 v hexadecimální soustavě, což je po převodu na číslo s pohyblivou řádovou čárkou 0,75.

Nejčastěji jsou používány tyto hodnoty:

| Zaslaná hodnota | Odpovídající hodnota |
|-----------------|----------------------|
| -1082130432 | 1 |
| -1086324736 | -0,75 |
| 1090519040 | -0,5 |
| -1098907648 | -0,25 |
| 0 | 0 |
| 1048576000 | 0,25 |
| 1056964608 | 0,5 |
| 1061158912 | 0,75 |
| 1065353216 | 1 |

Tabulka 2: Nejčastěji používané hodnoty pro ovládání dronu pomocí AT příkazů[14].

3.3 Zpracování řečového signálu

V minulé kapitole jsem popsal, jakými příkazy jsou drony ovládány, nicméně to pro samotné hlasové ovládání nestačí. Je nezbytné vědět, jakým způsobem se řečový signál zpracovává a jak probíhá samotné rozpoznávání slov. Hlavními zdroji pro kapitoly 3.3 až 3.6 jsou: [15], [16] a [17].

Proces zpracování řečového signálu pro účely rozpoznávání řeči je následující:

- Analogový signál z mikrofону je převeden na signál digitální.
- Digitální signál je segmentován na krátké úseky zvané segmenty.
- Segmenty jsou parametrizovány pomocí metod krátkodobé analýzy.
- Sled parametrizovaných segmentů se porovnává s modely slov.

3.3.1 Digitalizace

Pro digitalizaci zvukového signálu se používá pulzně kódové modulace (PCM). Probíhá ve třech krocích:

1. vzorkování
2. kvantizace
3. kódování

Vzorkování Vzorkování je proces, kdy se z časově spojitého signálu pravidelně odebírají vzorky v určených časových okamžicích. Tyto okamžiky jsou dány frekvencí, kterou označujeme jako vzorkovací frekvenci. Tato frekvence musí splňovat Shannon-Kotělníkův teorém, který vypadá následovně:

$$f_v > 2f_{max}, \quad (1)$$

kde f_v je vzorkovací frekvence a f_{max} je maximální frekvence vyskytující se v signálu.

Tento vztah nám říká, že pro jednu periodu původního signálu musíme odebrat alespoň dva vzorky. Pro lidskou řeč se používá vzorkovací frekvence 8 MHz. Při nedodržení Shannon-Kotělníkova teorému vzniká jev zvaný aliasing, ten se projevuje jako slyšitelné zkreslení v nízkých frekvencích.

Kvantizace Podstatou kvantizace je zaokrouhlení hodnot signálu získaného při vzorkování na předem definované kvantizační hladiny. Mezi nimi jsou rozhodovací stupně. Pokud je hodnota amplitudy signálu menší než rozhodovací stupeň, je kvantována pomocí nižšího kvantizačního stupně. V opačném případě je kvantována pomocí vyššího kvantizačního stupně. V průběhu kvantizace nastává kvantizační zkreslení, což je hodnota, o kterou jsme museli změnit původní amplitudu signálu.

Kódování Během kódování signálu je signál binárně kódován, což znamená, že se jednotlivým kvantizačním hladinám kvantovaného signálu přiřadí binární číslo.

3.3.2 Parametrizace

Úkolem parametrizace je potlačit vysokou informační redundanci řečového signálu. Redundance je způsobena tím, že každý člověk má originální hlasové ústrojí. Různí lidé mají tedy různé způsoby toho, jak vyslovují to samé slovo. Závisí to na způsobu artikulace, přízvuku, rychlosti řeči a dalších aspektech. Dokonce i stejný řečník může stejné slovo pokaždé vyslovit jinak. Záleží například na tom, zda řečník pokládá otázku nebo příkaz. To, jak je slovo vysloveno, může také ovlivnit například nálada nebo nemoc. Tyto rozdíly se projevují v délce jednotlivých úseků řeči a v intenzitě řečového signálu. Parametrizace je založena na poznatcích o tvorbě řeči, a sice že umíme rozlišit znělé hlásky, neznělé hlásky a mezery. Průběh parametrizace můžeme rozdělit na:

- Předzpracování
- Metody krátkodobé analýzy

3.3.3 Předzpracování

Aby bylo možné využít metod krátkodobé analýzy pro parametrizaci signálu, je třeba signál předzpracovat.

Předzpracování se obvykle skládá z následujících kroků:

- Segmentace
- Ustředění
- Preemfáze
- Váhování oken

3.3.4 Segmentace

Pro zpracování řečového signálu metodami krátkodobé analýzy je nutné, aby signál byl stacionární, což znamená, že musí mít stejné vlastnosti. Řečový signál však považujeme za náhodný, a tak je nutné jej rozdělit na dostatečně krátké časové úseky, ve kterých je možné signál považovat za stacionární. Na druhou stranu musí být délka úseků, které se nazývají segmenty, dostatečně velká pro přesnou analýzu signálu. V praxi se nejčastěji používá délka segmentu v rozmezí 10-30 ms. Segmenty se mohou částečně překrývat.

3.3.5 Ustředění

Ustředění se využívá, protože při snímání signálu může dojít k jeho stejnosměrnému posunutí v kladném či záporném směru. Toto posunutí může mít nepříznivý dopad na další zpracování, především pak na výpočty funkce krátkodobé energie. Ustředění jednoho rámce probíhá následovně:

$$s'(n) = s(n) - \frac{1}{N} \sum_{i=1}^N s(i) \quad (2)$$

pro $1 \leq n \leq N$,

kde $s'(n)$ je ustředěná hodnota, $s(n)$ je n -tá hodnota rámce, a N je počet vzorů obsažených v jednom rámci.

3.3.6 Preemfáze

Hlasové ústrojí produkuje řeč tak, že amplitudy harmonických složek klesají společně s frekvencí. Nicméně pro zpracování a následnou klasifikaci řeči je vhodnější, když je spektrum řeči rovnoměrné. Za tímto účelem se používá preemfáze, která funguje jako číslicový filtr zvýrazňující vyšší harmonické složky. Tento filtr lze popsat následující rovnicí:

$$y[n] = x[n] - \alpha x[n-1], \quad (3)$$

kde $y(n)$ je výstup filtru, $x(n)$ je vstupní vzorek filtru v čase n . Vstupní vzorek zpožděný o vzorkovací periodu se značí $x(n-1)$ a α je parametr. Parametr α bývá volen od 0.9 do 1.

3.3.7 Váhování oken

Jak už bylo zmíněno, pro krátkodobou analýzu je nutné zpracovávat signál po malých segmentech. Pro určitou velikost intervalu předpokládáme, že zvukový signál v jeho okolí je periodický s periodou uvnitř rámce. Není-li však perioda shodná s délkou segmentu, nebo pokud se segmenty překrývají, dopouštíme se jisté chyby ve zpracování. Nicméně tuto chybu můžeme částečně opravit použitím váhového okénka. Úkolem tohoto okénka je v analyzovaném segmentu vybrat vzorky signálu a při zpracování vyhladit jejich průběh přidělením určité váhy. Aplikace okénka na signál se prakticky provádí vynásobením odpovídající si hodnoty rámce a okna:

$$y(n) = x(n)w(n), \quad (4)$$

kde $y(n)$ je výstupní posloupnost, $x(n)$ je vstupní posloupnost a $w(n)$ je aplikované okno.

3.4 Metody krátkodobé analýzy

Při popisu parametrizace jsem zmínil, že jejím úkolem je potlačit redundanci v řečovém signálu. Toto mají za úkol metody krátkodobé analýzy. Snížení množství dat probíhá tak, že signál pro další zpracování reprezentujeme ne vlastními nasnímanými vzorky, ale souborem příznaků, které ze vzorku získáme. Snažíme se přitom, aby příznaky co nejlépe popisovaly vlastnosti signálu i jeho průběh. Metody krátkodobé analýzy se aplikují na jednotlivé segmenty získané při segmentaci. Pro každý segment získáme buď jeden příznak (skalár), anebo vektor příznaků, který popisuje signál v konkrétním segmentu.

Cílem krátkodobé analýzy je to, aby byl získaný popis jednoznačný a co nejméně ovlivněn okolními podmínkami. Musí tedy být co možná nejméně závislý na intenzitě signálu, na množství šumu v signálu a na řečnickovi. Metody krátkodobé analýzy je možné rozdělit na dvě kategorie. Dělíme je podle toho, zda vnímají spíše spektrální charakter signálu nebo s akustickým signálem pracují jako s časovou posloupností navzorkovaných hodnot.

3.4.1 Analýza v časové oblasti

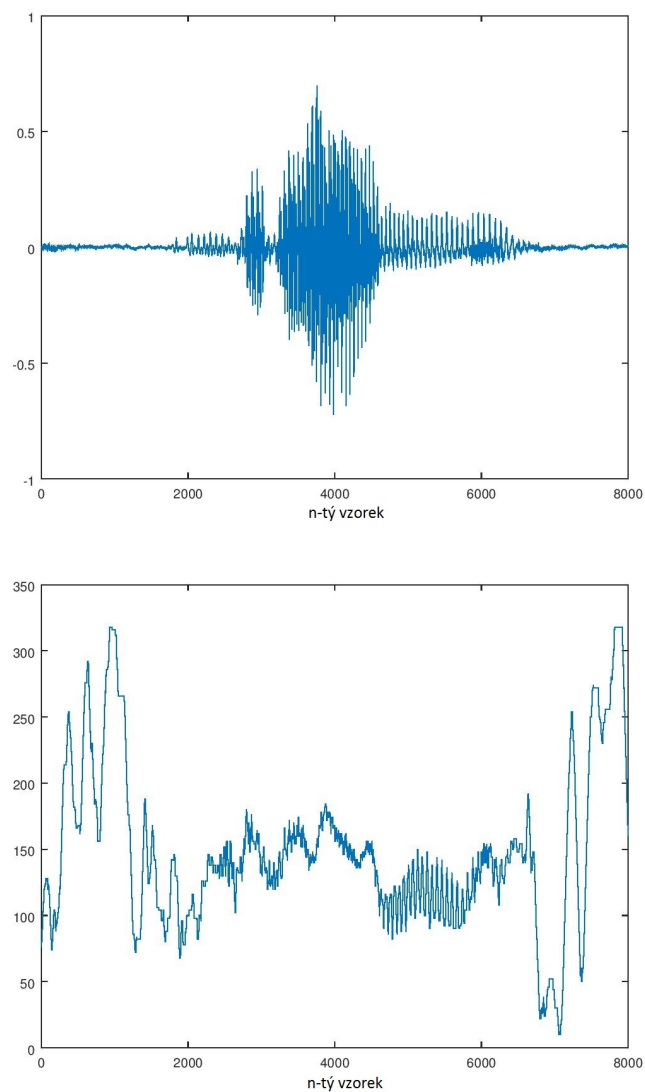
Při zpracování v časové oblasti vycházíme přímo z hodnot vzorků signálu. Příznaky, které získáme pomocí analýzy v časové oblasti, jsou většinou méně náročné na výpočet. Užitím jedné metody by pro mnohé fonémy hodnota příznaků zůstala stejná či se jen velmi málo změnila. Aby tedy bylo možné stanovit hranice všech fonémů, je vhodné použít více různých metod analýzy v časové oblasti a případně společně s nimi i metody pro analýzu ve frekvenční oblasti. Mezi tyto metody patří například počet průchodů nulou, krátkodobá energie a krátkodobá intenzita.

3.4.1.1 Počet průchodů nulou

Přestože tato metoda patří do metod pro analýzu časové oblasti, lze ji považovat za jednoduchou funkci popisující spektrální charakter signálu. Výhodou počtu průchodů nulou je úplná nezávislost na energii signálu. Pro získání počtu průchodů nulou se používá vztah:

$$Z = \sum_{k=0}^{N-1} | \operatorname{sgn}(x[k]) - \operatorname{sgn}(x[k-1]) | / 2, \quad (5)$$

kde $x[k]$ je vzorek signálu, k je pořadí vzorku v segmentu a N je počet vzorků v segmentu. Níže je pro ilustraci uveden graf.



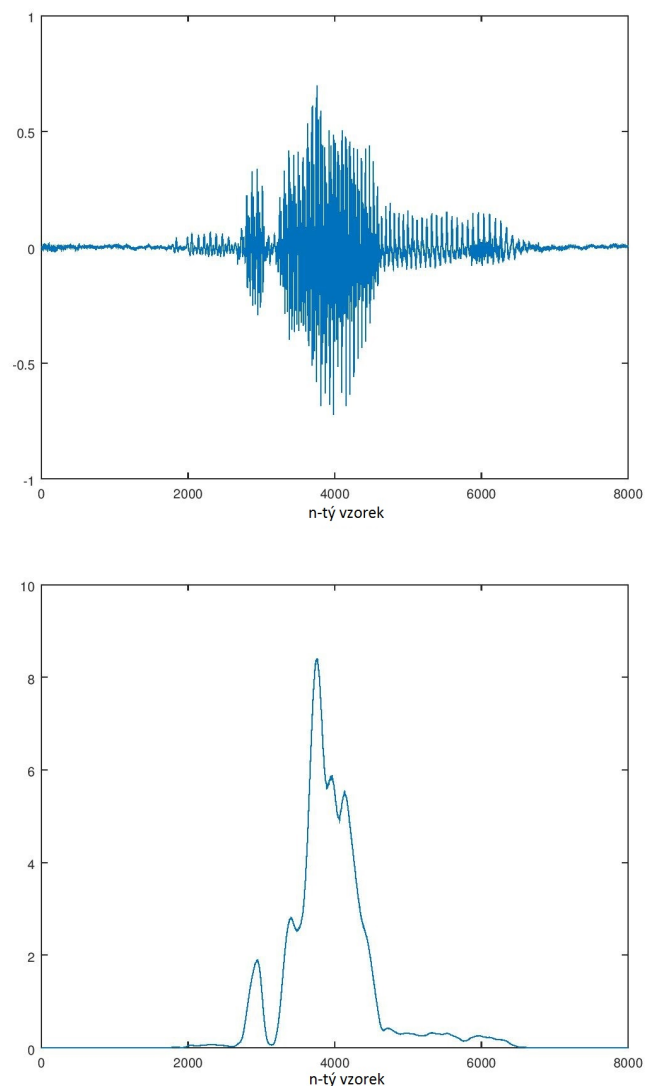
Obrázek 8: Signál slova dron a počet průchodů nulou v něm.

3.4.1.2 Krátkodobá energie

Tato metoda slouží pro výpočet energie segmentů. Pomocí ní lze určit hranice mezi znělými a neznělými úseky řeči. Výpočet krátkodobé energie lze definovat vztahem:

$$E = \sum_{k=0}^{N-1} x[k]^2, \quad (6)$$

kde $x[k]$ je vzorek signálu, k je pořadí vzorku v segmentu a N je počet vzorků v segmentu. Níže je pro ilustraci uveden graf.



Obrázek 9: Signál slova dron a průběh jeho krátkodobé energie.

3.4.1.3 Krátkodobá intenzita

Stejně jako krátkodobá energie slouží pro určení hranic znělých a neznělých úseků řeči. Nicméně oproti krátkodobé energii nezvýrazňuje tolik dynamiku řečového signálu (není zde umocněna). Výpočet krátkodobé intenzity lze definovat vztahem:

$$E = \sum_{k=0}^{N-1} x[k], \quad (7)$$

kde $x[k]$ je vzorek signálu, k je pořadí vzorku v segmentu a N je počet vzorků v segmentu.

3.4.2 Analýza ve frekvenční oblasti

Analýza ve frekvenční oblasti se také někdy nazývá krátkodobá spektrální analýza, protože řeč je ve frekvenční oblasti reprezentována svým spektrem. Spektrum se mění s tím, jak jsou pomocí hlasového ústrojí vydávány rozdílné zvuky. U těchto metod je výpočetní náročnost vyšší než u metod pro zpracování v časové oblasti. Nicméně často však oproti nim poskytují rozdílné výsledky, které by při použití metod pro analýzu v časové oblasti nebyly nalezeny. Metody pro analýzu ve frekvenční oblasti jsou nejčastěji založeny na použití krátkodobé Fourierovy transformace.

3.4.2.1 Krátkodobá diskrétní Fourierova transformace

Fourierova transformace slouží k převodu signálu z časové oblasti do frekvenční oblasti. Pro diskrétní signály využíváme diskrétní Fourierovu transformaci. Pomocí ní získáme spektrum signálu, ze kterého je možné určovat jeho kmitočtové vlastnosti. Fourierova transformace $S(\omega)$ posloupnosti $s(k)$ je dána vztahem:

$$S(\Omega) = \sum_{k=-\infty}^{\infty} S(k)e^{-i\Omega k} \quad (8)$$

3.4.2.2 Kepstrální analýza

Pomocí kepstrální analýzy můžeme zjistit všechny parametry pro model vytváření řeči. Těmito parametry jsou modifikace a buzení. Modifikace je dána hlasovým ústrojím. Buzení je dáno základním tónem řečníka, pro rozpoznání řeči jej nepotřebujeme a je tedy vhodné buzení oddělit od parametrů daných hlasovým traktem. Řečový signál můžeme považovat za konvoluci modifikace a buzení:

$$s(n) = x(n) * h(n), \quad (9)$$

kde $s(n)$ je řečový signál, $h(n)$ je modifikace a $x(n)$ je buzení. Nicméně ze signálu získaného konvolucí dvou vstupních signálů je obtížné dostat tyto signály zpět. Lze se o to však pokusit pomocí nelineární operace jako je například logaritmování. Tím je možné převést součin na součet, protože:

$$\ln(x * y) = \ln(x) + \ln(y) \quad (10)$$

Jednotlivé složky součtu pak lze od sebe oddělit.

Proces kepstrální analýzy tedy probíhá takto:

1. Provedeme diskrétní Fourierovu transformaci pro určení spektra analyzovaného segmentu.
2. Spektrum segmentu logaritmujeme.
3. Provedeme zpětnou(inverzní) diskrétní Fourierovu transformaci.

3.4.2.3 Melfrekvenční keprální koeficienty

U MFCC zohledňujeme fakt, že vnímání zvuku není lineární. Lidské ucho má totiž na nižších frekvencích větší rozlišení než na vysokých. Postup při MFCC je takový, že na frekvenční osu rozmístíme nelineárně trojúhelníkové filtry. Změříme jejich energii na výstupu a použijeme ji k výpočtu kepra místo diskrétní Fourierovy transformace.

Pro rozmístění filtrů na frekvenční ose využíváme převodu hertzů na mela dle následujícího vztahu:

$$F_{Mel} = 2959 \log_{10} \left(1 + \frac{F_{Hz}}{700} \right), \quad (11)$$

, kde F_{Mel} je frekvence v melech a F_{Hz} je frekvence v hertzech. Po tomto převodu můžeme umístit trojúhelníkové filtry lineárně na melové ose. Ty pak budou na standardní ose v hertzech rozmístěny nelineárně. Dále získáme energii trojúhelníkových filtrů, tu logaritmujeme. Nyní už jen musíme provést zpětnou Fourierovu transformaci. Využijeme diskrétní cosinové transformace, pomocí které získáme melfrekvenční keprální koeficienty. Obvykle se využívá 10 až 13 těchto koeficientů.

3.5 Rozpoznávání řeči

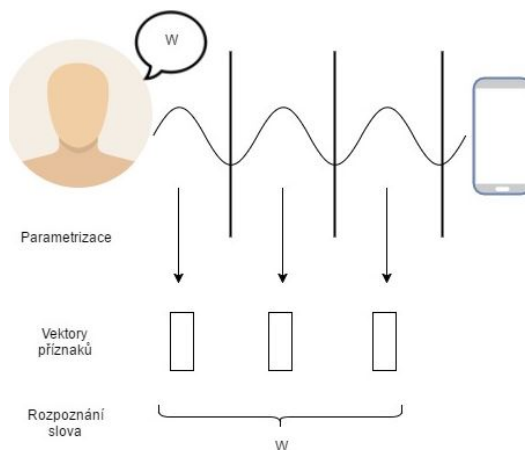
Rozpoznávání řeči má za úkol přiřadit pozorované promluvě informaci, která do ní byla vložena řečníkem při vyslovení.

Rozpoznávání řeči je možné rozdělit do tří kategorií:

1. **Rozpoznávání izolovaných slov** - jak už název napovídá, jedná se o rozpoznávání jednotlivých slov vyslovených izolovaně. Pro tyto účely se používá malý slovník.
2. **Rozpoznávání spojených slov** - zde se počítá s tím, že jednotlivá slova jsou vyslovována s pauzami. Pro rozpoznávání spojených slov je také dostačující malý slovník
3. **Rozpoznávání plynulé řeči** - plynulou řečí je myšleno to, že mezi jednotlivými slovy nejsou cílené krátké pauzy. Plynulá řeč se zpracovává hůře, protože je složitější najít začátek a konec slov. Dalším faktorem ovlivňujícím rozpoznání plynulé řeči může být například rychlost promluvy. Používají se velké slovníky pro tisíce až desetitisíce slov.

Další možné rozdělení rozpoznávání řeči je podle závislosti na konkrétním řečníkovi:

- **Rozpoznávání závislé na řečníkovi** - zde je možné porozumět pouze řečníkovi, který rozpoznávač natrénoval.
- **Rozpoznávání nezávislé na řečníkovi** - trénování rozpoznávače zde probíhá pomocí dat od velkého množství řečníků, což umožňuje nezávislost na konkrétním řečníkovi. Tento druh rozpoznávání řeči je dnes využíván nejvíce. Nejčastěji je založen na aplikaci skrytých Markovových modelů.



Obrázek 10: Jednoduché schéma rozpoznávání řeči.

3.5.1 Princip Markovova řetězce

Markovův řetězec popisuje diskrétní náhodný proces, kde platí, že pravděpodobnost přechodu z aktuálního stavu do stavu následujícího závisí pouze na stavu současném. To, jak se systém do současného stavu dostal, Markovův řetězec nepopisuje. Jednoduché zobrazení je znázorněno stavovým diagramem, který je zobrazen na obrázku níže. Ten popisuje, s jakou pravděpodobností se dostaneme ze současného uzlu do uzlu následujícího nebo jaká je pravděpodobnost, že v daném uzlu setrváme. Každé hraně je pak dána hodnota pravděpodobnosti, tak že součet hodnot hran vycházejících z jednoho uzlu musí být roven 1.



Obrázek 11: Stavový diagram Markovova řetězce.

3.5.2 Skrytý Markovův model

Skryté Markovovy modely[18] využívají princip Markovova řetězce. Pracují s výstupy algoritmů popsaných v kapitole Parametrizace, kterými jsou vektory příznaků popisující jednotlivé segmenty:

$$O = o_1, o_2, o_3 \dots o_T \quad (12)$$

jako O je celkový vektor slova, o_1 až o_T jsou vektory pozorování a T značí celkový počet pozorování a zároveň délku slova.

Rozpoznávací algoritmus musí vybrat ze slovníku slovo w_i , které je s největší pravděpodobností nejbližší zkoumanému slovu O . Tuto pravděpodobnost můžeme označit:

$$\operatorname{argmax} P(w_i|O) \quad (13)$$

Pro slovník platí, že:

$$W = w_1, w_2, \dots, w_m, \quad (14)$$

kde W je označení slovníku a w_1 až w_m jsou vzorové slova a m je jejich celkový počet. Tuto pravděpodobnost bohužel nelze vypočítat přímo, můžeme však použít Bayesova vztahu:

$$P(w_i|O) = \frac{P(O|w_i)P(w_i)}{P(O)} \quad (15)$$

Vztah vyjadřuje, že pravděpodobnost $P(O)$ je apriori, $P(w_i)$ je aposteriori. Pro optimalizaci je využit člen $P(O|W)$, což je věrohodnostní funkce. Tedy namísto toho, aby při pozorování O bylo hledáno nejvíce pravděpodobné w_i , hledáme nejvíce pravděpodobnou sekvenci vektorů O , která by mohla vygenerovat slovo w_i . Avšak zde nastává problém s rozměry dané sekvence O a přímém výpočtu podmíněné pravděpodobnosti, který není prakticky realizovatelný.

Nahradíme-li však parametrický model slova jiným typem modelu, je problém řešitelný. Tímto jiným typem modelu je skrytý Markovův model. Namísto toho, abychom prováděli odhad pravděpodobnosti $P(O|W)$, budeme odhadovat pravděpodobnost $P(O|W) = P(O|\lambda_i)$, kde λ_i jsou jednotlivé parametry skrytého Markovova modelu. Předpokládáme tedy, že každé vzorové slovo w_i reprezentuje skrytý Markovův model o parametrech λ_i .

Existuje několik typů Markovových modelů. V oblasti rozpoznávání řeči však používáme nejčastěji levo-pravý Markovův model.

Matematicky je skrytý Markovův model popsán maticí přechodových pravděpodobností A . Její prvky a_{ij} udávají pravděpodobnost přechodu ze stavu i do stavu j . Dále je model popsán vektorem počátečních pravděpodobností π , což jsou pravděpodobnosti, se kterými se bude proces vyskytovat na počátku generování ve stavu daném indexem vektoru.

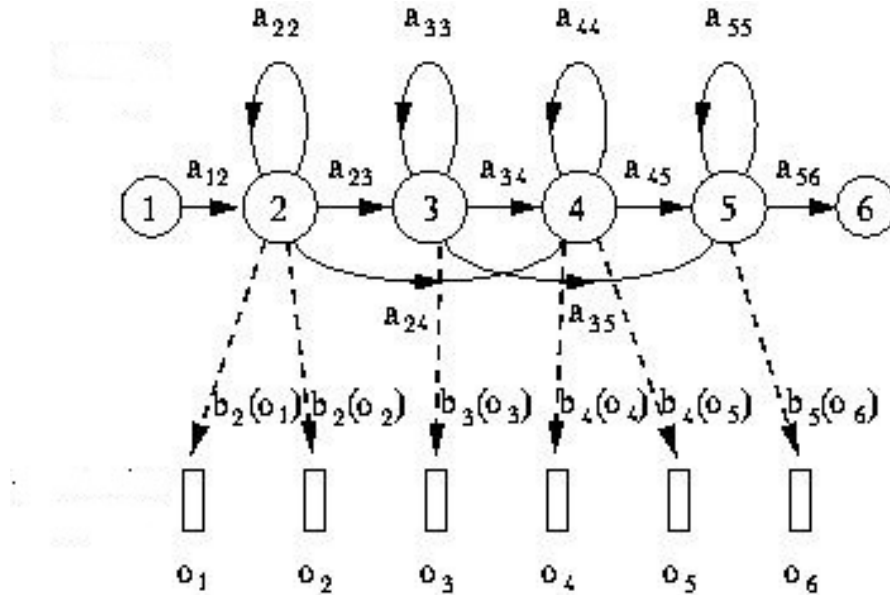
Poslední součástí je rozdělení výstupní pravděpodobností generování vektorů pozorování stavy $b_j(o_t)$. Pro prvky matice přechodových pravděpodobností platí, že pro $j < i$ jsou rovny nule. Nemůžeme totiž postupovat proti časové ose, protože, jak už bylo zmíněno, pravděpodob-

nost přechodu z aktuálního stavu do stavu následujícího závisí v Markovových modelech pouze na aktuálním stavu. Nenulové prvky matice přechodových pravděpodobností tedy leží pouze na hlavní diagonále a nad ní.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} \quad (16)$$

$$\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{pmatrix} \quad (17)$$

Na následujícím obrázku můžeme vidět levo-pravý Markovův model o šesti stavech, z nichž jen čtyři (2,3,4,5) jsou emitující stavy, to znamená, že generují pozorování a funkce rozdělení výstupní pravděpodobnosti. Zároveň tyto stavy odpovídají jednotlivým fonémům. Řekněme, že tento model je modelem slova word. Fonému o tedy náleží stav 3, přechodové pravděpodobnosti a_{33} a a_{34} a funkce rozdělení výstupní pravděpodobnosti b_3 .



Obrázek 12: Levo-pravý Markovův model o šesti stavech[18].

3.6 Dynamické borcení časové osy

Dynamické borcení časové osy (DTW), je starší metoda, která je ale stále využívána pro svou jednoduchost. Tato metoda se využívá hlavně při rozpoznávání izolovaných slov.

DTW při porovnávání dvou slov určuje vzdálenost promluv, tedy testovací sekvence O a referenční sekvence R .

$$O = o_1, o_2, o_3 \dots o_O \quad (18)$$

$$R = r_1, r_2, r_3 \dots r_R \quad (19)$$

kde $r(i)$ a $o(j)$ jsou vektory se stejným počtem prvků. Pro měření vzdáleností vektorů by nám postačila Euklidovská metrika, nicméně slova nemají pouze jeden vektor, ale sekvenci vektorů. Problémem také je, že slova nejsou nikdy stejně dlouhá, ani pokud je vysloví stejný řečník. To se dá vyřešit pomocí transformační funkce času, která jedno slovo "natáhne" na slovo druhé. Výpočet vzdálenosti dvou sekvencí pomocí transformační funkce času je dána vztahem:

$$D_c(O, R) = \frac{\sum d(r(r_c(k)), o(o_c(k))) W_c(k)}{N_c}, \quad (20)$$

kde $r_c(i)$ je testovací sekvence a $o_c(i)$ je sekvence referenční, c je cesta, N_c je normalizační faktor a $W_c(k)$ je váha, odpovídající k -tému kroku cesty.

Vzdálenost sekvencí O a R je pak dána jako minimální vzdálenost přes soubor všech možných cest:

$$D(O, R) = \min_{\{C\}} D_C(O, R) \quad (21)$$

Výpočet vzdálenosti DTW je možno vypočítat následovně:

Nejprve musíme získat **matici vzájemných vzdáleností**. Rozměry této matice jsou $R * O$ kde R je počet vektorů referenční sekvence a O je počet vektorů testovací sekvence. Jednotlivé prvky matice se spočítají pomocí vztahu $D_{ij} = d(r(i), o(j))$ kde d je metrika pro výpočet vzdáleností dvou vektorů, například výše uvedená Euklidovská metrika.

Pokud máme sestavenou matici vzájemných vzdáleností můžeme přistoupit k sestavení **matic kumulovaných vzdáleností**, kterou budeme značit G . Platí, že:

$$G_{0,0} = 0 \quad \text{a} \quad G(0, m \neq 0) = G(n \neq 0, 0) = \infty \quad (22)$$

Jednotlivé prvky matice G pak spočítáme takto:

$$G_{m,n} = \min \left\{ \begin{array}{l} G_{m-1,n} + D_{m,n} \\ G_{m-1,n-1} + 2x D_{m,n} \\ G_{m,n-1} + D_{m,n} \end{array} \right\} \quad (23)$$

Konkrétní příklad pak může vypadat takto s tím, že referenční sekvence má 4 vektory a testovací pouze 3. Předpokládejme, že jsme získali matici vzájemných vzdáleností D a z ní podle výše uvedeného vztahu získáme prvky matice kumulovaných vzdáleností G .

$$D = \begin{vmatrix} 1 & 5 & 3 \\ 5 & 2 & 1 \\ 4 & 3 & 6 \\ 2 & 2 & 5 \end{vmatrix} \quad (24)$$

$$G = \begin{vmatrix} \infty & 14 & 16 & 14 \\ \infty & 13 & 11 & 11 \\ \infty & 8 & 9 & 15 \\ \infty & 4 & 6 & 11 \\ 0 & \infty & \infty & \infty \end{vmatrix} \quad (25)$$

Na následujícím obrázku je naznačená cesta:

$$G = \begin{vmatrix} \infty & 14 & 16 & 14 \\ \infty & 13 & 11 & 11 \\ \infty & 8 & 9 & 15 \\ \infty & 4 & 6 & 11 \\ 0 & \infty & \infty & \infty \end{vmatrix}$$

Obrázek 13: Naznačená cesta v matici G .

nyní už můžeme jednoduše určit vzdálenost tím, že hodnotu prvku na konci cesty podělíme součtem počtů vektorů testovacích a referenčních.

$$D_{DTW} = \frac{14}{3 + 4} = 2 \quad (26)$$

3.6.1 Rozpoznávání pomocí DTW

Od rozpoznávání očekáváme správné přiřazení promluvy k určitému slovu ve slovníku. Slovník obsahuje slovní třídy, přičemž každá tato třída zastupuje jedno slovo. V těchto třídách jsou pak uloženy promluvy daného slova od jednoho či více řečníků. Pomocí DTW vypočítáme jednotlivé vzdálenosti promluvy a referencí ze slovních tříd. Pokud máme referencí více, je třeba nejprve určit vzdálenost celé slovní třídy od testované promluvy, a až poté je možné určit, o jaké slovo se jedná.

Toto můžeme řešit dvěma metodami:

1. 1-NN neboli nearest neighbor. Vzálenost se určí podle minimální vzdálenosti reference v slovní třídě.

2. k-NN neboli k nearest neighbors. Zde probíhá výpočet tak, že si nejdříve seřadíme jednotlivé vzdálenosti referencí od nejmenší po největší a vzdálenost se určí pomocí průměrné hodnoty k nejmenších hodnot.

Když máme vypočteny vzdálenosti jednotlivých slovních tříd od dané promluvy, můžeme přejít k rozhodování, o které slovo se jedná. K tomu slouží dvě metody:

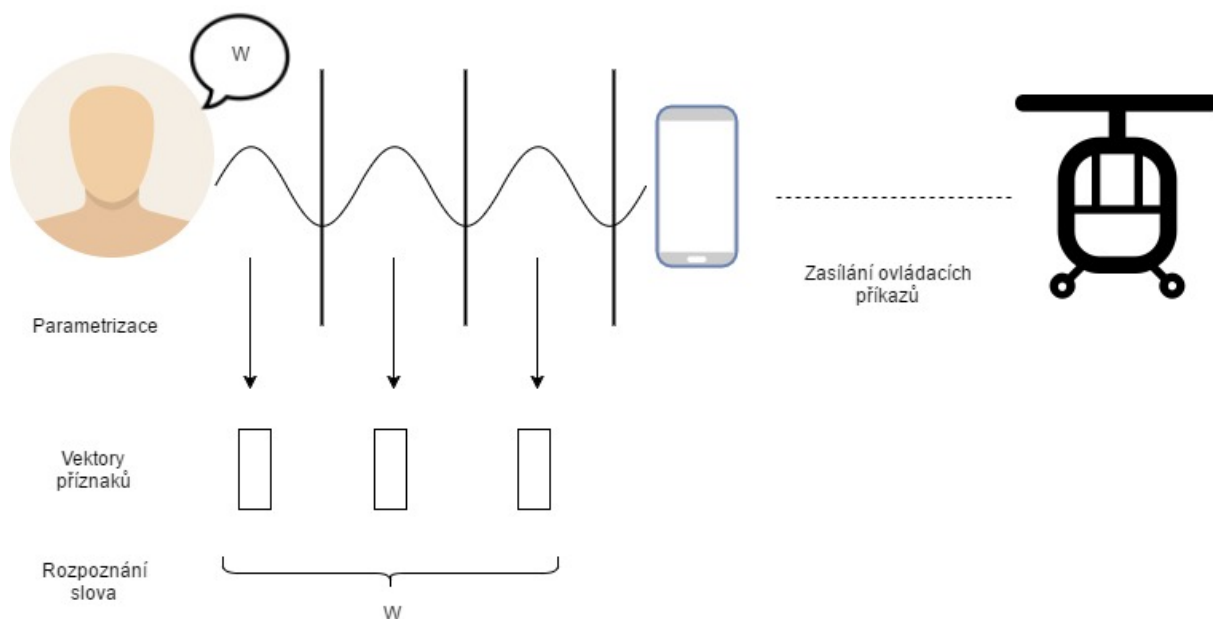
1. Minimální vzdálenost, kde je rozhodnuto, o které slovo se jedná na základě nejmenší vzdálenosti mezi slovní třídou a promluvou.
2. Prahová hodnota, kde je rozhodnuto, o které slovo se jedná na základě prahové hodnoty. Když vzdálenost klesne pod tuto hodnotu, má se za to, že daná slovní třída reprezentuje testovanou promluvu. U této metody však může nastat i situace kdy je jedné testovací promluvě, přiřazeno více referenčních slov.

3.7 Další možnosti rozpoznávání řeči

Jako další využitelnou možnost při rozpoznávání řeči bych rád uvedl rozpoznávání řeči v systému Android. Pro programování v tomto systému je možné využít RecognizerIntent, který nám umožňuje přistupovat k rozpoznávání řeči v systému Android, jenž je založen na neuronových sítích. Výsledek tohoto intentu je možné reprezentovat pomocí proměnné String, která může být následně použita v podmínkách. Pokud jsou tyto podmínky splněny, je možné provádět různé úkony, mezi které by mohlo patřit i zaslání ovládacího příkazu do dronu. Rozpoznávání řeči od společnosti Google podporuje nejen angličtinu, ale i velké množství dalších jazyků, mezi které patří i čeština[19].

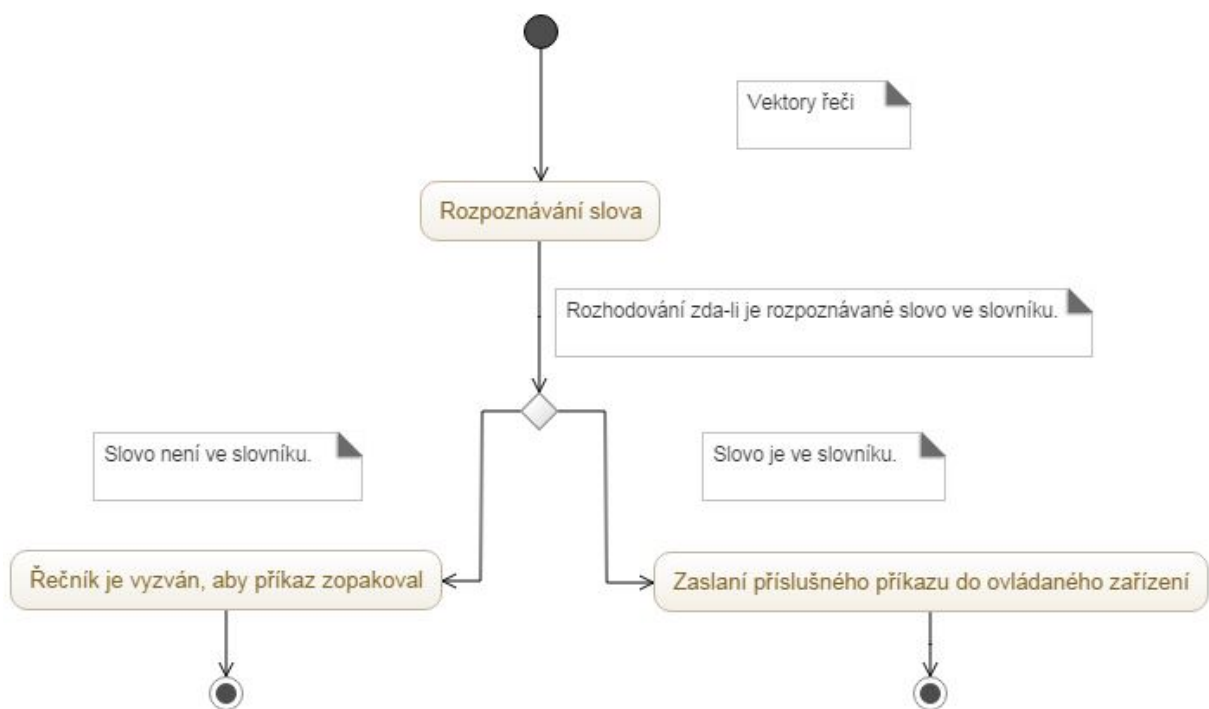
3.8 Návrh komunikačního schématu mezi mobilním zařízením a dronem

Nyní již víme, jak probíhá zpracování řečového signálu, rozpoznávání řeči a jakým způsobem jsou dronu posílány ovládací příkazy. Můžeme tedy přejít k návrhu celého komunikačního schématu prezentovaného na obrázku níže:



Obrázek 14: Návrh komunikačního schématu mezi mobilním zařízením a dronem.

Promluva řečníka je nejprve zaznamenána mikrofonom a následně je analogový signál z mikrofону digitalizován. Dalším krokem je segmentace, při které rozdělíme signál na krátké segmenty stejné délky v rozmezí 10-30 ms. Následuje parametrizace, kde na každý segment aplikujeme některou z metod krátkodobé analýzy, přičemž v praxi se nejvíce používají Melfrekvenční cepstrální koeficienty. Výstupem těchto metod jsou vektory příznaků. Dalším krokem je pak samotné rozpoznávání slova, tedy porovnání vektorů příznaků s jednotlivými modely slov, které náš slovník obsahuje. O shodě bude rozhodovat jedna z metod použitých výše, tedy využijeme buď Skrytých Markovových modelů nebo Dynamického borcení časové osy. Je nutné zohlednit i fakt, že promluva nemusí být vždy rozpoznána, proto se rozpoznávání bude řídit podle následujícího diagramu:



Obrázek 15: Diagram aktivit pro rozpoznávání řeči.

Ze schématu je patrné, že pokud slovo rozpoznáno nebude, bude řečník vyzván, aby promluvu zopakoval, případně aby zadal jiný příkaz. Pokud však promluva bude rozpoznána, bude se pokračovat dle výše uvedeného schématu. Dle rozpoznání slova se do dronu odešle zpráva obsahující příslušný příkaz a ten pak bude dronem proveden.

4 Teoretický návrh hlasových příkazů

Nyní již máme teoretický návrh komunikačního schématu a zbývá definovat hlasové příkazy. Jak bylo výše zmíněno, v praxi by se hlasové ovládání dalo uplatnit například u gimbalu. Co to je gimbal bylo již popsáno v úvodu této práce, nicméně pro ilustraci zde uvádím obrázek obrázek⁴ konkrétního gimbalu:



Obrázek 16: Gimbal Mini 3D Pro Feiyu Tech

Do gimbalu se připevní kamera nebo fotoaparát. Nejčastěji se používají kamery od společnosti GoPro, například GoPro Hero 5. Tento model zde uvádím mimo jiné proto, že má implementováno hlasové ovládání. To je možné provádět v několika různých jazycích. Mezi příkazy patří například nahrávání videa a ukončení nahrávání, pořízení snímku, pořízení sekvence snímků a další. Kameru lze ke gimbalu většinou připojit pomocí micro hdmi kabelu, mezi tyto kamery patří i GoPro Hero 5. Existují ale také kamery, které toto neumožňují. Pro tyto kamery se využívá bezdrátová manuální spoušť, která se ke kameře připojuje pomocí jacků.

Dále je vhodné se zaměřit na řídicí jednotky gimbalů. Ačkoli existuje velké množství těchto jednotek, rozhodl jsem se zaměřit na jednotku Storm 32[20]. Zvolil jsem ji z několika důvodů. Kromě toho, že umožňuje kontrolovat pohyb gimbalu po všech třech osách, je použita v řadě projektů, mezi kterými je i open source aplikace pro ovládání přes mobilní telefon. Tato aplikace by případně mohla být rozšířena o hlasové ovládání a gimbal by tak mohl být hlasem ovládán pomocí mobilního telefonu. Pro Storm 32 existuje také projekt vytvořený pro sledování určitého GPS bodu, což je důležité pro níže zmíněný příkaz follow. Další výhodou je to, že tato řídicí jednotka umožňuje komunikaci pomocí Micro Air Vehicle Link protokolu (zkráceně MAVLink).

⁴<https://www.megapixel.cz/feiyu-tech-stabilizator-mini-3d-pro>

Pomocí tohoto protokolu je možné provádět komunikaci s mikropočítači, jako je například Arduino nebo Raspberry Pi, které je možné využít pro ovládání dronu.

4.1 Příkazy pro ovládání gimbalu

Níže si popíšeme příkazy, které jsou pro ovládání gimbalu vhodné. Nejdříve je nutno definovat příkazy pro ovládání pohybu gimbalu:

- Start - Příkaz pro inicializaci hlasového ovládání.
- Left – Po přijetí příkazu se začne gimbal natáčet doleva rychlostí $3^\circ/s$.
- Right – Po přijetí příkazu se začne gimbal natáčet doprava rychlostí $3^\circ/s$.
- Up – Po přijetí příkazu se začne gimbal natáčet nahoru rychlostí $3^\circ/s$.
- Down – Po přijetí příkazu se začne gimbal natáčet dolů rychlostí $3^\circ/s$.
- Forward – Po přijetí příkazu bude gimbal stále směřovat dopředu, bez ohledu na to jakým směrem se bude pohybovat dron.
- Bottom - Po přijetí příkazu bude gimbal stále směřovat kolmo dolů, bez ohledu na to jakým směrem se bude pohybovat dron.
- Follow – Po přijetí příkazu bude gimbal stále směřovat na určený GPS bod, bez ohledu na to jakým směrem se bude pohybovat dron.

Dále je nutné definovat příkazy pro nahrávání videa a pořizování snímků:

- Video – Tento příkaz spustí nahrávání videa.
- Foto – Tento příkaz spustí sekvenci snímků s rychlostí snímek/s.
- Shot – Po přijetí příkazu se pořídí snímek.

Posledním potřebným příkazem je příkaz Stop, který ukončí poslední probíhající příkaz. Tento příkaz je velmi důležitý nejen pro ukončování nahrávání videa a sekvencí snímků, ale také proto, že veškeré hlasové příkazy, u kterých není pevně definována doba trvání, je třeba nějak zastavit. Pro představu, pokud by gimbal byl ovládán klasickým pákovým ovladačem, tak by se otáčel daným směrem pouze po dobu, kdy je páka daným směrem vychýlena. Když však zadáme hlasový příkaz, tak dron provádí příkaz kontinuálně, a je tedy nutné jej, v případě potřeby, ukončit.

Toto byly příkazy pro obecné ovládání gimbalu, ale níže uvedu několik příkladů praktického využití dronů, u kterých by se dalo využít hlasového ovládání, a také uvedu konkrétní hlasové příkazy pro dané využití.

4.2 Konkrétní příkazy dle způsobu využití dronu

4.2.1 Chytré zemědělství

V této oblasti se drony využívají zejména k získávání informací o zdraví pěstovaných rostlin, oblastech kde mají rostliny nedostatek živin či vody, nebo také k měření a hodnocení rozsahu škod po živelných katastrofách. Aby toto bylo možné, musí být dron vybaven lidarem. Kromě lidarů se používá také kamera pro pořizování snímků a za účelem počítání rostlin, respektive odhadu jejich počtu. Pro tento způsob využití dronu je určitě možné využít hlasového ovládání, pokud k obecnému ovládání gimbalu přidáme ještě několik příkazů. Jednodušší a univerzálnější než využívat konkrétní názvy funkcí bude využívat příkazy ve formátu function one až function x. Takto definované příkazy pak ulehčí využití funkcí i u jiných případů užití.

- Function one – Tento příkaz spustí funkci, která hodnotí zdraví rostlin na základě dat získaných lidarem.
- Function two – Tento příkaz spustí funkci, která ukáže oblasti, ve kterých mají rostliny nedostatek živin nebo vody, na základě dat získaných lidarem.
- Function three – Tento příkaz spustí funkci, která vyhodnotí rozsah škody po zasažení živelnou katastrofou.
- Function four - Tento příkaz spustí funkci, která odhadne počet rostlin v dané oblasti.
- Mode lidar - Tento příkaz způsobí, že dron bude využívat zařízení lidar
- Mode camera - Tento příkaz způsobí, že dron bude využívat kameru

4.2.2 Kontrola potrubí

Drony se v této oblasti využívají zejména pro detekci termálních anomálií pomocí termokamery. Dále se pak drony využívají také kvůli vizuální inspekci. Hlasové ovládání je zde vhodné obohatit o přepínání režimu zobrazení a zoom pro vizuální inspekci.

- Zoom - Po přijetí tohoto příkazu začne kamera přibližovat.
- Mode thermal - Tento příkaz způsobí, že dron bude využívat termokameru za účelem inspekce termálních anomálií.
- Mode camera - Tento příkaz způsobí, že dron bude využívat kameru za účelem vizuální inspekce potrubí.

4.2.3 Solární elektrárny

Dále se drony využívají pro inspekci solárních elektráren. Nejen pro vizuální inspekci, ale v dnešní době umí drony také detekovat nefunkční nebo prasklý solární panel. V tomto případě je tedy vhodné doplnit hlasové ovládání o příkaz zoom, navíc je také vhodné definovat příkaz pro periodický pohyb, kdy bude dron například obepisovat kružnici o rádiu 10 metrů.

- Zoom – Po přijetí tohoto příkazu začne kamera přibližovat.
- Circle x – Po přijetí tohoto příkazu začne dron obepisovat kružnici o rádiu x, přičemž x bude celé číslo vyjádřeno v metrech.

4.2.4 Těžební průmysl

V těžebním průmyslu se drony využívají pro odhad množství sypkého nevytěženého materiálu v kubických metrech v povrchových dolech. Zde je možné definovat periodickou funkci, která způsobí, že bude dron obepisovat kružnici o určitém rádiu nad povrchovým dolem, a příkaz zoom.

- Zoom – Po přijetí tohoto příkazu začne kamera přibližovat.
- Circle x – Po přijetí tohoto příkazu začne dron obepisovat kružnici o rádiu x, přičemž x bude celé číslo vyjádřeno v metrech.

4.2.5 Integrovaný záchranný systém a armáda

Využití dronů v této oblasti je spjato převážně s využitím termokamery za účelem detekce osob. Využívají se však také speciální čočky, které umožňují noční vidění. Je tedy vhodné definovat příkazy pro přepínání módů snímání, ale také příkazy pro periodickou funkci a zoom.

- Zoom – Po přijetí tohoto příkazu začne kamera přibližovat.
- Circle x – Po přijetí tohoto příkazu začne dron obepisovat kružnici o rádiu x, přičemž x bude celé číslo vyjádřeno v metrech.
- Mode thermal - Tento příkaz způsobí, že dron bude využívat termokameru, za účelem detekce osob.
- Mode camera - Tento příkaz způsobí, že dron bude využívat kameru, případně kameru s čočkou pro noční vidění.

4.2.6 Hnojení pomocí dronů

V oblasti hnojení byl společností DJI vyvinut dron AGRAS MG-1. Tento dron sice nevyužívá gimbal, nicméně pořád by zde bylo vhodné navrhnout příkaz pro periodickou funkci.

- Circle x – Po přijetí tohoto příkazu začne dron obepisovat kružnici o rádiu x , přičemž x bude celé číslo vyjádřeno v metrech.

5 Předpoklady pro možnou praktickou implementaci a specifikace možných problémů

Výstupem všech předchozích kapitol je teoretický návrh, ať už komunikačního schématu nebo hlasových příkazů. Je tedy nutné popsat předpoklady, které musí být splněny, aby tyto teoretické návrhy bylo možné uskutečnit.

Nejprve je třeba zmínit, že pro konkrétní návrh zpráv, které budou zasílány dronu, je nutné znát formát těchto zpráv daného výrobce, protože pro konkrétní výrobce a zařízení se tento formát může lišit. Dalším bodem je pak vytvoření konkrétního slovníku, který musí obsahovat všechny slova použitá pro hlasové ovládání. To může být časově náročná operace, obzvláště pokud bychom chtěli, aby hlasové ovládání nebylo závislé na řečníkovi, protože by bylo třeba získat co nejvíce referenčních promluv od většího počtu řečníků. Dále je nutné zohlednit fakt, že odezva při hlasovém ovládání bude větší než při ovládání pomocí ovladače.

Vhodné je pak také zmínit výhody a nevýhody hlasového ovládání.

Výhody:

- Nutnost menšího počtu lidí pro ovládání profesionálních dronů.
- Provádění periodických úkonů. Jako příklad si můžeme představit, že dronem monitorujeme nějaký komplex budov. Pak je předpoklad, že budeme opakovaně využívat určitý postup či dráhu letu, přičemž pro tento účel by bylo možné definovat příkaz, který by pak například gimbal natáčel automaticky a člověk ovládající dron by už jen kontroloval, aby dráha dronu vyhovovala přednastavenému natáčení gimbalu.

Nevýhody:

- Špatná využitelnost hlasového ovládání v oblastech se zvýšeným ruchem.
- Problém rozpoznávání řeči u osob s vadou řeči. Nicméně se dá předpokládat, že pro hlasové ovládání profesionálního dronu by byl vybrán člověk, který vadou řeči netrpí.
- V dnešní době se dá za nevýhodu považovat i vysoká finanční náročnost.
- Poslední nevýhodou je, že oproti pákovému ovládání je potřeba příkaz buď nějak ukončit a nebo definovat dobu jeho vykonávání. Toto může být nicméně vyřešeno rozpoznáváním spojených slov, kdy by řečník nejprve řekl příkaz, a po krátké pauze by bylo řečeno, jakou dobu má být příkaz vykonáván.

6 Závěr

Cílem této bakalářské práce bylo navrhnout komunikační schéma mezi uživatelem a bezpilotním dronem, dále definovat jednotlivé hlasové příkazy a popsat algoritmus detekce a přiřazení správného příkazu k hlasovému vstupu.

Nejprve tato práce zpracovává informace o dronech a jejich ovládání. Následně byla provedena rešerše stávajících řešení pro hlasové ovládání bezpilotních modelů, která zjistila, že v momentální době neexistuje funkční řešení pro hlasové ovládání periferií dronu.

Dále se práce zabývá popisem metod pro zpracovávání řečového signálu a problematikou rozpoznávání slov. Na základě těchto informací bylo navrženo komunikační schéma mezi uživatelem a bezpilotním dronem.

Na toto navazuje vlastní návrh hlasových příkazů. Ten je rozdělen na dvě části, z nichž se první zabývá příkazy pro ovládání gimbalu, kde jsou definovány příkazy pro základní pohyb a funkce gimbalu, a druhá se věnuje konkrétním příkazům dle způsobu využití dronu a jsou zde definovány příkazy pro speciální funkce, periodický pohyb, či příkazy pro změnu aktuálně používaného zařízení.

Poslední kapitola pojednává o předpokladech pro možnou praktickou implementaci a specifikaci možných problémů, a také o možných výhodách a nevýhodách zavedení hlasového ovládání. V návaznosti na tuto práci je možné vypracovat konkrétní praktické řešení na základě komunikačního schématu a hlasových příkazů navržených v této práci. S tím, že podmínky pro tuto praktickou implementaci shrnuje poslední kapitola této práce.

7 Seznam použité literatury

Literatura

- [1] The UAV. The UAV [online]. [cit. 2017-03-09]. Dostupné z: <http://www.theuav.com/>
- [2] KARAS, Jakub a Tomáš TICHÝ. Drony. Brno: Computer Press, 2016. ISBN 978-80-251-4680-4.
- [3] Sciencedirect.com. (2017). Modeling and Backstepping-based Nonlinear Control Strategy for a 6 DOF Quadrotor Helicopter - ScienceDirect. [online] Dostupné z: <http://www.sciencedirect.com/science/article/pii/S1000936108600345> [cit. 2017-03-09].
- [4] UAV Coach. (2017). How to Fly a Quadcopter - The Ultimate Guide | UAV Coach. [online] Dostupné z: <http://uavcoach.com/how-to-fly-a-quadcopter-guide/> [cit. 2017-03-09].
- [5] Voice Controlled Drone with RasPi, Amazon Echo and 3DR IRIS+. Hackster.io [online]. 2016 Dostupné z: <https://www.hackster.io/veggiebenz/voice-controlled-drone-with-raspi-amazon-echo-and-3dr-iris-c9fd2a> [cit. 2017-03-14].
- [6] Node-bebop library [online]. Dostupné z: <https://github.com/hybridgroup/node-bebop> [cit. 2017-03-14].
- [7] How to Build a Portable Voice Controlled Drone for Under \$500. Mobiquity.com [online]. Dostupné z: <https://www.mobiquityinc.com/innovation/how-to-build-a-portable-voice-controlled-drone-for-under-500> [cit. 2017-03-14].
- [8] Perceptual Drone Speech Recognition. Intel Developer Zone [online]. 2015 Dostupné z: <https://software.intel.com/en-us/articles/perceptual-drone-speech-recognition-using-realsense> [cit. 2017-03-14].
- [9] Voice Controlled Arduino Drone. Instructables [online]. Dostupné z: <http://www.instructables.com/id/SpeechVoice-Controlled-Arduino-Drone/> [cit. 2017-03-14].

- [10] Parrot BEBOP 2. Parrot [online]. Dostupné z: [https:// www.parrot.com/ ca/ drones/ parrot-bebop-2](https://www.parrot.com/ca/drones/parrot-bebop-2) [cit. 2017-03-14].
- [11] PARROT SA. ARSDK Protocols [online]. 2015 Dostupné z: http://developer.parrot.com/docs/bebop/ARSDK_Protocols.pdf [cit. 2017-03-14].
- [12] CMU Sphinx. [online]. Dostupné z: <http://cmusphinx.sourceforge.net/wiki/> [cit. 2017-03-14].
- [13] Anon, (2017). [online] Dostupné z: https://homes.cs.washington.edu/~shwetak/classes/ee472/notes/ARDrone_SDK_1_6_Developer_Guide.pdf [cit. 2017-03-09].
- [14] Robotappstore.com. (2017). Robot App Store | Knowledge-Base | How to Control ARDrone Movements During Flight. [online] Available Dostupné z: <http://www.robotappstore.com/Knowledge-Base/How-to-Control-ARDrone-Movements-During-Flight/99.html> [cit. 2017-03-09].
- [15] PSUTKA, Josef. Mluvíme s počítačem česky. Praha: Academia, 2006. Česká matice technická (Academia). ISBN 80-200-1309-1.
- [16] Anon, (2017). [online] Dostupné z: http://www.fit.vutbr.cz/study/courses/ZRE/public/opora/zre_opora.pdf [cit. 2017-03-09].
- [17] Anon, (2017). [online] Dostupné z: http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/digitalspeechprocessingcourse/final_speech_paper_1_2008.pdf [cit. 2017-03-09].
- [18] Ee.columbia.edu. (2017). HTKBook. [online] Dostupné z: <http://www.ee.columbia.edu/ln/LabROSA/doc/HTKBook21/HTKBook.html> [cit. 2017-03-09].
- [19] Developer.android.com. (2017). RecognizerIntent | Android Developers. [online] Dostupné z: <https://developer.android.com/reference/android/speech/RecognizerIntent.html> [cit. 2017-03-09].

- [20] Olliw.eu. (2017). STorM32-BGC Wiki. [online] Available at: http://www.olliw.eu/storm32bgc-wiki/Main_Page [Accessed 17 Apr. 2017].
- [21] HOHENLOHE, Stephan zu. Drony: stručně a přehledně: výběr vhodného modelu, ovládání, foto a video, legislativa. Přeložil Richard KŘÍŽ. Frýdek-Místek: Alpress, 2016. ISBN 9788075432346.
- [22] DJI Official. (2017). DJI Phantom Drone. [online] Available Dostupné z: <http://www.dji.com/phantom> [cit. 2017-03-28].
- [23] Finecosz.com. (2017). Fineco Science & Technology Co., Ltd. [online] Dostupné z: http://www.finecosz.com/products_detail/productId=68.html [cit. 2017-03-09].
- [24] Community.parrot.com. (2017). Bebop 2 Knowledge Base - Parrot. [online] Dostupné z: https://community.parrot.com/t5/tkb/allarticlesprintpage/tkb-id/Bebop_2@tkb [cit. 2017-03-09].
- [25] Megapixel, s. (2017). Feiyu Tech stabilizátor Mini 3D Pro. [online] Megapixel.cz. Dostupné z: <https://www.megapixel.cz/feiyu-tech-stabilizator-mini-3d-pro> [cit. 2017-03-09].